CHAPTER

5

HARDWARE AND DEVICE MANAGEMENT IN WINDOWS 2000

After reading this chapter and completing the exercises, you will be able to:

- Differentiate between hardware and devices
- Describe how the operating system manages devices and resource allocation
- Examine how devices communicate with the CPU and other system resources
- List the four components of hardware management in Windows 2000 and describe how they interact
- Describe how the new Win32 Driver Model (WDM) fits into the Windows 2000 architecture
- Understand Windows 2000 support for Power Management and the OnNow system
- Describe Windows 2000 support for hardware devices and system bases
- Demonstrate the device and power management configuration tools within Windows 2000
- Troubleshoot Windows 2000 device drivers and hardware

Hardware management is an important facet of any operating system—and Windows 2000 is no exception. The ability to smoothly and efficiently handle a large number of hardware devices, from that old 14.4 Kbps modem to the latest DVD disc player, is one of the shining achievements of Windows 2000. As we look into the internals of Windows 2000, you will see that Microsoft has built in a large amount of hardware support that device manufacturers once provided via very complex device drivers. Hardware management in Windows 2000 is divided into four distinct yet interdependent components: **Win32 Driver Model (WDM)**, **power management**, **hardware devices**, and **system buses**. We will examine the design and operation of each of these components in detail as we take an overall view of how hardware and device management is handled in Windows 2000.

An Overview of Hardware Management

Our first task is to distinguish between hardware and devices. As shown in Figure 5-1, hardware is a physical component (or peripheral) and a device is a generic operating system term for the entire stack of software that supports and maintains the hardware. For example, you can install hardware such as a VGA display adapter in a Windows 2000 computer. The supporting drivers for that hardware present a device with standardized application programming interfaces (APIs), I/O ports, and memory ranges available for use by the operating system. As such, device drivers provide an important avenue of communications between the physical hardware and the various pieces of the operating system.

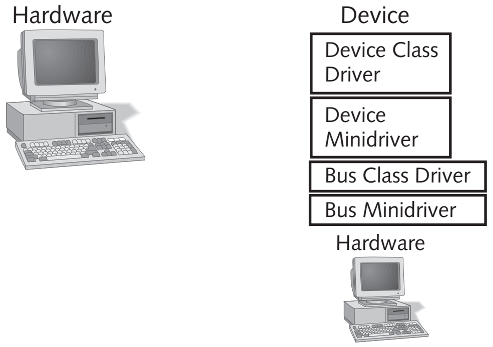


Figure 5-1 A device consists of hardware and supporting device drivers needed for use by the operating system

With the release of Windows 98 and Windows 2000, Microsoft radically changed the internal operations and design of device drivers. The new device driver design is called the WDM. Although the new WDM concepts are quite similar to both the **VxD driver** concepts used in Windows 95 and the existing "NT-style" device drivers used in Windows NT 4.0/3.51, there are significant implementation differences between these driver technologies. In particular, installation and device detection is handled is quite differently between the older driver specifications and the new WDM model. The new WDM architecture supports the **Plug and Play** standard, whereas the older Windows NT-style drivers do not. Although most Windows NT- and VxD-style device drivers operate satisfactorily under Windows 2000, a wealth of new features and capabilities are available to the enterprising developer who embraces the WDM architecture.

One of the most important benefits of the WDM design is that, in most cases, device drivers written under WDM can be deployed without any modification in both Windows 98 and Windows 2000 environments. This standardization should significantly ease the development burden for device manufacturers and driver developers.

A primary advantage of the original Windows NT operating system is the hardware abstraction layer (HAL), which acts as a buffer to keep ill-behaved applications from crashing the entire operating system by accessing the hardware directly. HAL is still around in Windows 2000. Applications continue to pass hardware calls through the driver stack to the HAL, which then passes those requests to the hardware itself (theoretically), protecting the operating system from application-induced crashes. This approach of buffering the hardware from direct application access works very well in Windows 2000, even if it's not quite foolproof. Accommodations must be made in the device driver coding, however, to account for the fact that the HAL protects devices from direct access by the operating system components and application software running on the machine.

The new WDM architecture also includes a layered approach to device drivers, which provides some of the base functionality in built-in WDM class drivers that ship with Windows 98 and Windows 2000. This layered approach means that for broad classes of standard hardware and peripheral devices [like hard drives or **Universal Serial Bus (USB)** devices], only a small piece of code, called a minidriver, needs to be developed by device manufacturers. Microsoft has already provided the many basic driver capabilities in the standard Windows 2000 bus class drivers.

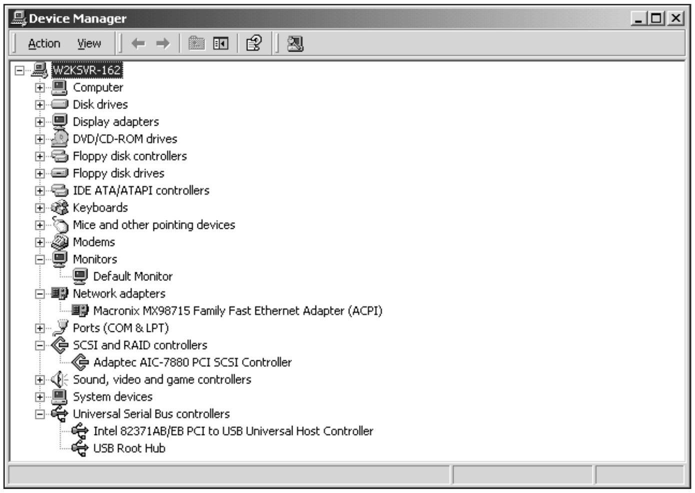
Windows 2000 includes extensive built-in support for advanced power management features, such as *suspend* and *resume*, for every piece of system hardware and all attached peripherals. This expansion of power management capabilities is part of the PC industry's efforts to make PCs as easy to use and as quick to turn on as the average toaster. The only way to accomplish such a feat with a PC is to keep the computer and all peripherals in a low-power suspend mode, which we like to call "the big sleep." Much like the Monty Python "Dead Parrot" sketch, the PC isn't really dead, it's just resting.

In addition, the Windows 2000 architecture supports multiple system buses and a wide variety of devices in an effort to give the user total hardware management. The system buses include **IEEE 1394 Serial Bus (FireWire)** and USB support native to the operating system. Windows 2000 also offers direct support for many newfangled technologies, such as DVD (Digital Versitile Disk), imaging, multiple display adapters, and digital audio devices.

The following sections discuss, in considerable detail, hardware and device management in Windows 2000. They then explain how you can apply this in-depth knowledge to understand and troubleshoot hardware problems in Windows 2000.

HARDWARE AND DEVICES: WHICH IS WHICH?

What's the difference between hardware and devices? As this chapter was being written, that very question kept popping up again and again. After reviewing numerous Microsoft Web sites and documentation, we finally have an intelligent answer that attempts to differentiate between the two terms. Hardware means just that: a physical piece of computer equipment, such as a motherboard, modem, or peripheral. Windows 2000 communicates with and manages hardware via defined devices. A device is a combination of hardware and the supporting software—device drivers, class drivers, minidrivers, and so on—used to make the hardware available for use by the operating system. If you peruse the Windows 2000 **Device Manager** (Start, Settings, Control Panel, System, choose the Hardware tab, and click the Device Manager button), you will see a list of all available devices, as shown in Figure 5-2. If you double-click a particular device, you can read the driver details for the software supporting that device. You can think of devices as a package of physical hardware and the software used to make that hardware useful to the operating system. In reality, the two terms are frequently used interchangeably.



```
W2KSVR-162
  Computer
  Disk drives
  Display adapters
  DVD/CD-ROM drives
  Floppy disk controllers
  Floppy disk drives
  IDE ATA/ATAPI controllers
  Keyboards
  Mice and other pointing devices
  Modems
  Monitors
    Default Monitor
  Network adapters
    Macronix MX98715 Family Fast Ethernet Adapter (ACPI)
  Ports (COM & LPT)
  SCSI and RAID controllers
    Adaptec AIC-7880 PCI SCSI Controller
  Sound, video and game controllers
  System devices
  Universal Serial Bus controllers
    Intel 82371AB/EB PCI to USB Universal Host Controller
    USB Root Hub
```

Figure 5-2 The Device Manager provides a list of all system hardware

How Windows 2000 Manages Devices and Resource Allocation

During the past five years, the proliferation of adapters using the **PCI bus** standard and the popularity of Plug and Play hardware has greatly altered the way in which an operating system manages devices. The original motivation for the PCI bus was twofold: to provide for software configuration (that is, no jumpers to set on the boards) of hardware adapters, and to increase the clock speed and throughput of the primary system bus in PCs. The impetus for Plug and Play was to allow the computer to intelligently choose a harmonious configuration for all compliant devices in a PC. As hardware is added or removed from a personal computer, the **Plug and Play Manager** automatically updates the system configuration, resource allocation, and device drivers to reflect the device changes.

The process by which Windows 2000 discovers which devices are installed and which resources are being used by those devices is called **enumeration**. In enumeration, each installed device is polled and a hardware tree is built in the Registry to reflect the machine's configuration. The Plug and Play Manager provides the arbitration process for resource allocation, and enumeration is the method by which Windows 2000 discovers the installed devices and resources in use. After enumeration is complete, the Plug and Play Manager modifies the Registry by creating a device tree on a bus-by-bus basis. This device tree includes a branch for each device, the unique device identification, a list of resources required by the device, and a list of resources allocated to the device. You can view the current hardware tree in the Control Panel, System applet by clicking the Device Manager button on the Hardware tab. Figure 5-3 shows one such Device Manager hardware tree.

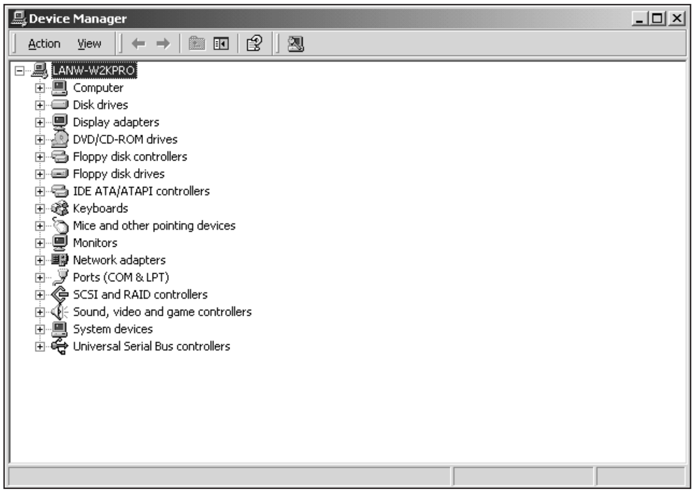


Figure 5-3 Enumeration creates a hardware tree in the Registry, which can be displayed in the Device Manager

Another important part of the enumeration process is the loading of device drivers. Once again, the Plug and Play Manager has a role. It dynamically loads and unloads the appropriate device drivers based on the findings of the enumeration process. Because of the highly evolved functionality of Plug and Play adapters and the **Advanced Configuration and Power Interface** (**ACPI) specification**, little or no user configuration is required when adding new devices to your Windows 2000 system.

How Windows 2000 Communicates with the CPU and Other System Resources

We are now ready to take a quick look at the actual path that Windows 2000 uses in communciations involving the operating system, system devices, and memory. The following sections go into some detail about how hardware and devices are recognized and managed under Windows 2000. Figure 5-4 shows a simplified rendition of the various communication pathways between devices, device drivers, and system resources, such as RAM and the CPU.

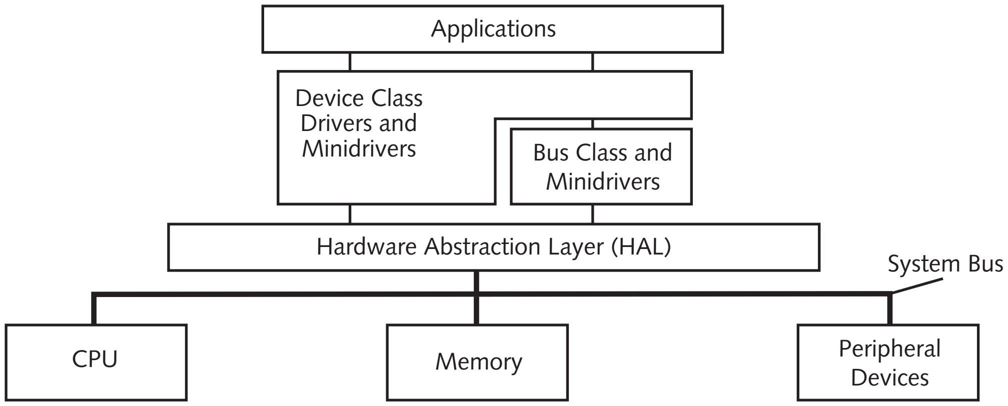

Applications
Device Class Drivers and Minidrivers
Bus Class and Minidrivers
Hardware Abstraction Layer (HAL)
System Bus
CPU
Memory
Peripheral Devices

Figure 5-4 The data path among system components is dependent on the system bus

THE FOUR COMPONENTS OF HARDWARE MANAGEMENT

As mentioned earlier, the four primary components of hardware management implemented in Windows 2000 are WDM layers, power management, hardware devices, and system buses. Each hardware management component provides specific services and capabilities within the operating system. Although we discuss each of these components as if they are separate and distinct, in reality these components are very tightly integrated into the operating system—and into each other—to provide a comprehensive method for managing hardware. WDM is the glue that holds the overall hardware management components together. It also provides an exciting new standard method for implementing device drivers, moving a large portion of device communications into standard class drivers provided with the operating system and thereby greatly easing the burden of driver developers.

WIN32 DRIVER MODEL AND WINDOWS 2000 ARCHITECTURE

The new WDM device driver specification represents a major leap in device management and driver design. Traditionally, device drivers had to take into account myriad details and functions to manage installation, communication, and maintenance between hardware and the underlying operating system. The development effort required was quite large because all device drivers had to control and track all user configuration information, hardware communication, API calls, and any required services. By transferring a large number of the low-level device management to standardized Windows 2000 class drivers, Microsoft has enabled faster device driver development, greater standardization in the interaction between drivers and the operating system, and increased usability because WDM is supported by both Windows 98 and Windows 2000.

The WDM Layered Driver Approach

The WDM architecture allows for a layered approach in which the majority of the layers for most devices are supplied by the Windows 2000 operating system. Figure 5-5 illustrates an example of the layered approach to device drivers implemented in the WDM. Your actual layers might differ according to the configuration of your particular machine.

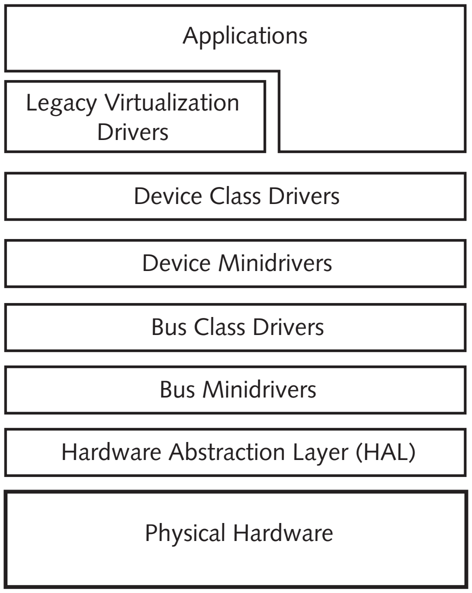


Figure 5-5 WDM layers include class drivers and Device Minidrivers

Applications communicate either with the legacy virtualization layer (discussed later) or directly with a **device class driver**. Device class drivers are standard drivers provided by WDM. A device class driver talks to a **device minidriver**, which is provided by each device manufacturer or driver developer and is specific to a particular piece of hardware. Depending on the hardware supported, the device minidriver might pass instructions and information to the **bus class driver**, which would communicate with the **bus minidriver**, which would in turn communicate with the HAL, as shown in Figure 5-5.

Class and Bus Drivers Built into Windows 2000

Hardware devices supported directly under the WDM specification include the following:

- Power management
- USB
- Plug and Play
- Digital audio
- Imaging (still image and video capture)

- Human Interface Device
- IEEE 1394 Serial Bus (FireWire)
- DVD

Note

If you are developing a device driver for hardware in one of these categories, a minidriver is all that's required in most instances. WDM and Windows 2000 handle the majority of the driver functions through either built-in class or bus drivers, or both, depending on your specific device requirements.

If, for example, you were writing a driver to provide support for a FireWire-based keyboard device under Windows 2000, you would have to develop a device minidriver to interact with the human interface device class *and* a bus minidriver to interact with the IEEE 1394 bus class driver of WDM. API support and resource management are built into the device class and bus class drivers, and development of the two minidrivers to support your new FireWire keyboard will be reasonably simple. Compared with older device driver techniques, in which developers included sophisticated code for management of resources, communications, and device housekeeping, minidrivers are a snap. In fact, the Windows 2000 class and bus drivers often provide enough support to enable all of the standard tasks and resources for a particular device. For example, a standard VGA adapter operates quite well with the built-in class.

The Legacy Virtualization Drivers Layer of WDM Although the WDM device drivers are the preferred method of implementing hardware support in Windows 2000, older VxD-style drivers are also supported in the new architecture so as to maintain some backward compatibility with existing hardware and software. The **legacy virtualization drivers layer** provides that compatibility support. Looking back at Figure 5-5, you can see this layer near the top of the **driver stack**, just under the Applications layer. The legacy virtualization drivers provide support for older VxD drivers by intercepting API and functions calls that, for example, might access a mouse or other input device hardware directly in flight simulation software running in a **Virtual DOS Machine** of Windows 98. As the application tries to access hardware via the VxD driver, the legacy virtualization drivers intercept those requests and map them to the appropriate class or bus drivers in Windows 2000. In the case of the flight simulator example, user input requests would be mapped to the **human interface device class**, which supports hardware, such as keyboards, mice, virtual reality gloves, and joysticks.

The legacy virtualization drivers layer provides another important service to VxD-style device drivers: it acts as a legacy device "traffic cop." Some legacy device drivers are accustomed to taking full control of any hardware used by the driver and accessing that hardware directly by design. The legacy virtualization drivers layer provides a seamless method of allowing multiple VxD-style device drivers to share the same hardware or software without conflicts. This device time-sharing process is accomplished by monitoring and maintaining the proper state of hardware devices as the various drivers gain and relinquish control through the legacy virtualization drivers layer. This task is much more difficult than it appears at first glance, because different drivers expect the same device to be in different states of operation. For example, if device driver 1 is accumulating mouse input and device driver 2 interrupts that operation for its own input purposes, the virtualization drivers must fool driver 1 into believing that the device is still operating under the control of driver 1. While driver 2 has control of the mouse, the virtualization drivers maintain the state of the mouse hardware expected by driver 1. After driver 2 completes its use of the mouse input, the virtualization drivers restore the mouse to the proper state necessary for driver 1 to continue using the mouse without failure or timeout of the device driver.

POWER MANAGEMENT AND THE ONNOW SYSTEM

Windows 2000 provides power management through the integrated **OnNow system**, which governs the power states and requirements for all aspects of a Windows 2000-based PC. OnNow works in conjunction with an Advanced Configuration and Power Interface (ACPI)-enabled BIOS to gather motherboard and BIOS information regarding power management and Plug and Play device configuration. This granular approach to power management allows OnNow to put individual devices within a system into a reduced-power mode while they are not actively being used. This innate ability to turn devices on as needed and to put them to "sleep" when idle requires some changes to the operating system and applications. The developers of many popular applications will undoubtedly modify their products to conform to the OnNow specification as Windows 2000 gains acceptance. Figure 5-6 shows the architecture of the OnNow system components.

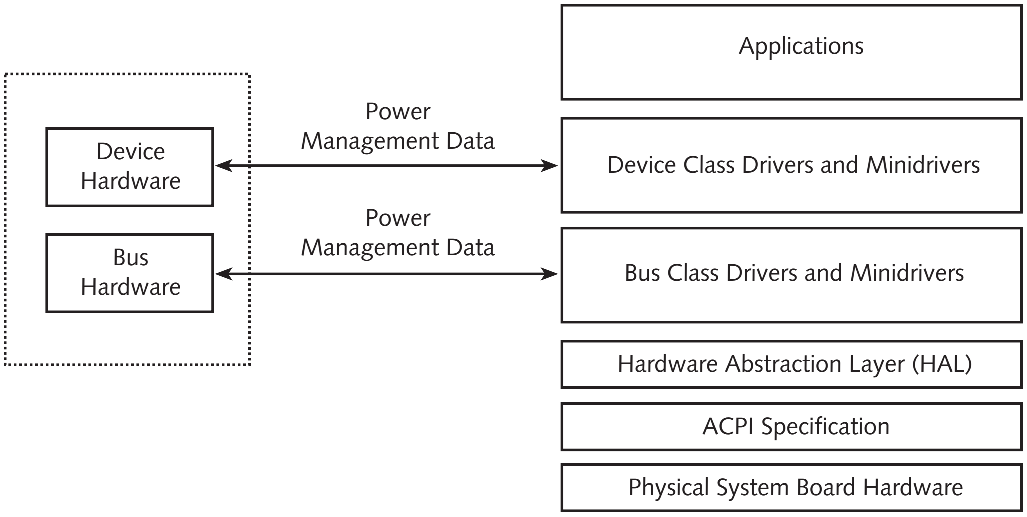

Applications
Power Management Data
Device Hardware
Device Class Drivers and Minidrivers
Power Management Data
Bus Hardware
Bus Class Drivers and Minidrivers
Hardware Abstraction Layer (HAL)
ACPI Specification
Physical System Board Hardware

Figure 5-6 The OnNow system defines a power management data path between the driver stack and the physical hardware

Many current applications assume that all attached system devices and peripherals are always up and running. Consequently, applications such as fax software or backup software will need to be rewritten to add OnNow support for sophisticated device power management. For example, applications running under OnNow must take into account that an idle device might be automatically put into sleep mode by Windows 2000 and that OnNow will need some time to "wake up" the hardware before the device can be accessed by the application. This latency characteristic would crash most current applications that access hardware on a regular basis. WDM supports OnNow by defining specific driver interfaces called **device driver interfaces (DDIs)**, which can detect idle services and handle power state changes at the device level.

HARDWARE DEVICES

The third component of the Windows 2000 operating system is hardware devices. As mentioned earlier in our discussion of device and bus class drivers built into Windows 2000, several broad categories of hardware devices are supported directly by Windows 2000 and WDM:

- human interface device
- Multiple display support
- Video capture
- Still image capture
- Digital audio
- DVD
- AGP (Accelerated Graphics Port)

Our earlier discussion of these device classes considered them in relation to the WDM device class and minidriver model. Now let's look at each class of hardware device in more detail.

The human interface device class represents the new standard method by which input devices interact with Windows 2000. It includes such input devices as a mouse and other pointing devices, game controllers, keyboards, and virtual reality controllers. How will device class drivers and bus class drivers interact for most hardware devices? The majority of devices supported under the human interface device class will use the new USB bus as an input interface into the PC. Hence, a USB-based human interface device input device will require the use of both the human interface device class driver and the USB bus class driver.

Multiple display support is now native to the operating system with Windows 2000. Although various manufacturers have implemented proprietary multiple display support for several years, Windows 2000 now implements this feature as an integral part of the operating system. You can define as many as nine monitors easily under Windows 2000 and specify the starting-point and end-point coordinates for each monitor. Figure 5-7 shows a typical multiple monitor setup under Windows 2000.

Configuration of multiple displays is handled through the Control Panel, Display applet (Start, Settings, Control Panel, Display, choose the Settings tab, then click Identify). You may be thinking, "How in the world can I install and configure nine display adapters in my PC?" Don't forget that the FireWire bus can support 127 devices, and many display manufacturers are including a FireWire interface in their new monitors.

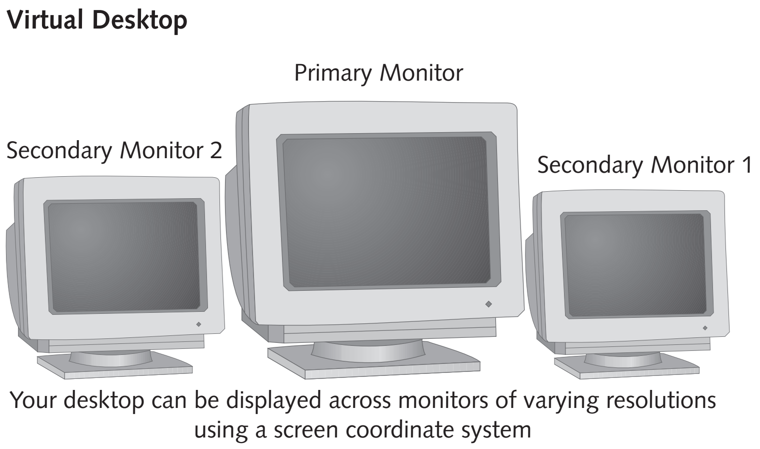

Virtual Desktop
Primary Monitor
Secondary Monitor 2
Secondary Monitor 1
Your desktop can be displayed across monitors of varying resolutions using a screen coordinate system

Figure 5-7 Win2K supports as many as nine displays using a coordinates system to map the virtual desktop

SYSTEM BUSES

A bus is simply a communication pipeline between various hardware components of a PC. Windows 2000 includes native support for four system buses:

- PCI bus
- PC Card and CardBus
- IEEE 1394 bus (FireWire)
- USB bus

Each bus serves a different, distinct purpose in the PC. For example, the **CardBus** supports laptop add-in cards in a manner similar to the older PCMCIA (Personal Computer Memory Card International Association) specification, but at much greater throughput and with enhanced Plug and Play capabilities. This new-specification laptop adapter is referred to as a **PC Card**. The PCI bus is now the standard high-performance bus used to connect system memory, the CPU, system cache, add-in adapters, and other system buses. FireWire is a relatively new serial bus standard that allows much greater throughput, hot-swap capabilities, device daisy-chaining, and easier device configuration than the older RS-232 serial standard. USB is also a new serial bus standard, although USB and FireWire are optimized for different classes of devices.

FireWire supports as many as 63 devices per bus and is designed for efficient connection of high-bandwidth consumer electronics, such as digital, still, and video cameras, high-speed scanners and printers, and DVD drives. You can also interconnect as many as 1023 FireWire buses to support a massive total of more than 64,000 devices. That should be enough for even the most ardent computer gadget freak! Figure 5-8 shows an example of a typical FireWire bus. Note that multiple computers can share a FireWire device, such as a printer, by using a FireWire splitter. Obviously, the printer in this example is designed for shared access from multiple computers.

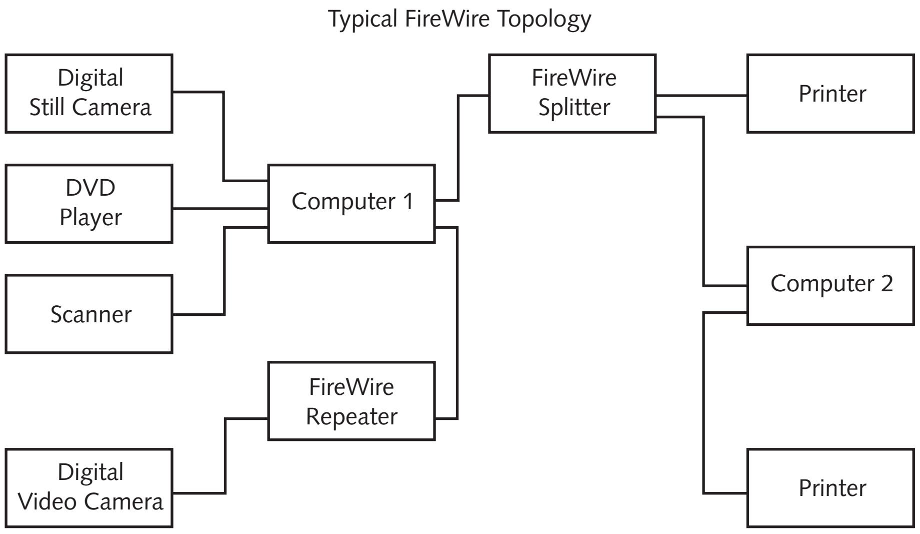


Figure 5-8 The FireWire bus supports repeaters and splitters

USB serves as a single port connection that can support multiple system input devices, such as a mouse, keyboards, joysticks, and digital audio playback, all on the same bus. This bus supports 127 devices in a tiered topology. Figure 5-9 shows an example of a typical USB topology.

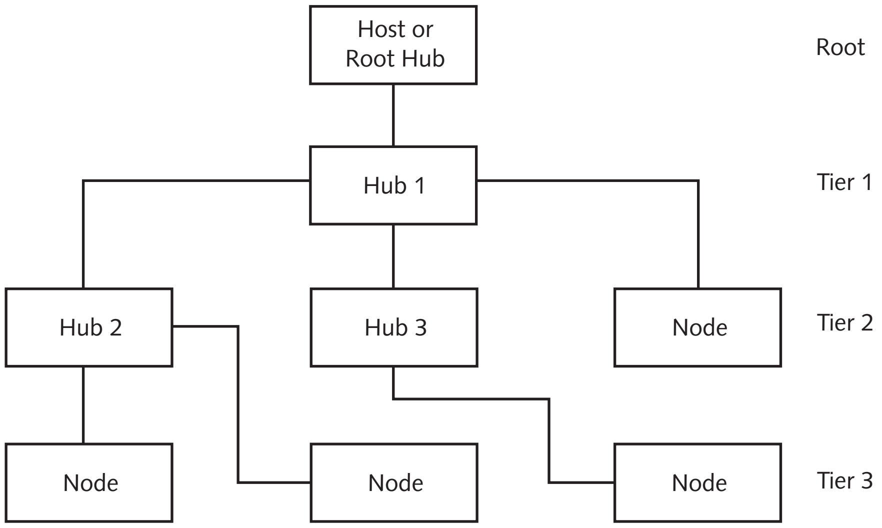


Figure 5-9 The USB bus features a tiered topology

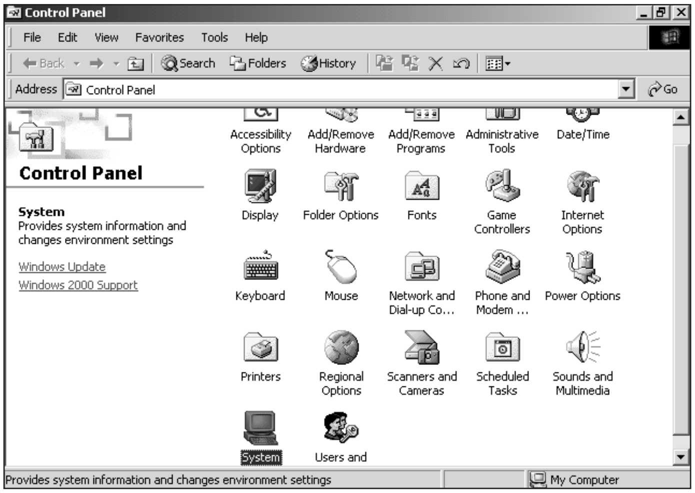
Note

Even though we have not mentioned support for some other very popular PC system buses (such as ISA, EISA, and SCSI), Windows 2000 does accommodate devices utilizing these buses, as well. These buses are not directly and natively supported along with the other bus class drivers primarily because of architectural issues and the need for manual configuration of adapters using these buses.

By now, you can probably see a pattern on the part of Microsoft—it has tried to make installation and configuration of hardware as easy as those processes are under the Apple Macintosh architecture. Windows 2000 and Windows 98 clearly emphasize technologies such as Plug and Play because of their ease of use for operating system users. Older technologies and buses are still supported, but not as directly or as elegantly as the new technologies are. Hardware manufacturers have been served notice by Microsoft where to direct their new product development efforts and are being actively enticed by the WDM architecture. Microsoft knows that the best way to speed adoption of new technologies is to make them very cost-effective for the manufacturer while showing end users the clear benefits of upgrading.

CONFIGURING USER INPUT, DISPLAY DEVICES, AND POWER MANAGEMENT

Even the best-laid plans of the Plug and Play Manager can sometimes go awry. Where do you turn when you want to change your device's configuration? How do you adjust your display settings or calibrate an input device? The Control Panel is the right place to start. See Figure 5-10.



Control Panel

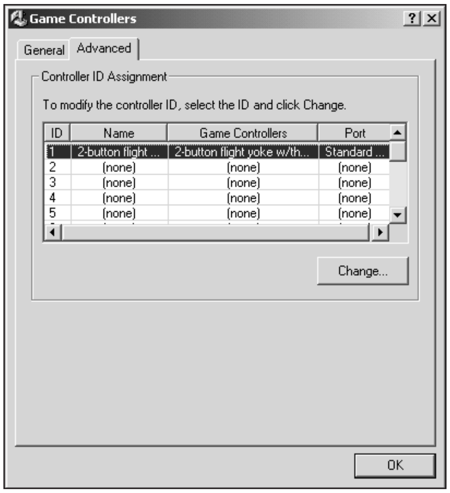
System
Provides system information and changes environment settings

Provides system information and changes environment settings

Figure 5-10 The Control Panel is the starting place for most device and operating system configuration actions

Configuring Display and Input Devices

When you look at the various standard applets under the Control Panel in Windows 2000, you will certainly notice the Display applet. Using this applet, you can perform myriad display configuration tasks, such as changing the system display resolution, configuring a screen saver, configuring the power management options of your monitor, and changing the appearance of your Windows desktop. For configuration of input devices, you have several pertinent applets, depending on the type of input device. For instance, the Game Controllers applet configures joysticks, and flight simulation yokes, as shown in Figure 5-11.

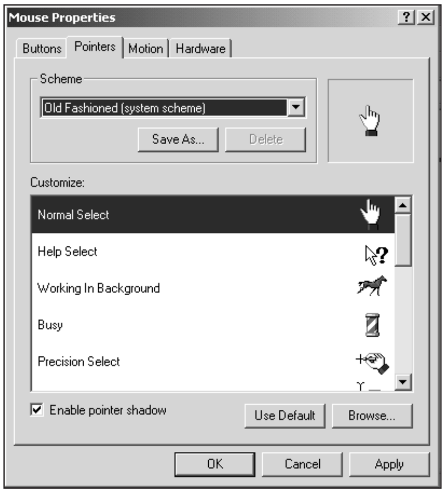


To modify the controller ID, select the ID and click Change.

ID	Name	Game Controllers	Port
1	2-button flight ...	2-button flight yoke w/th...	Standard ...
2	(none)	(none)	(none)
3	(none)	(none)	(none)
4	(none)	(none)	(none)
5	(none)	(none)	(none)

Figure 5-11 The Game Controllers applet controls configuration and calibration of game input devices

The Keyboard applet allows you to configure the typematic rate—how quickly characters are repeated when you hold a key down, the rate of blinking for the system cursor, and the language and layout of the keyboard. The Mouse applet is one of the most used of the input device applets. It allows you to configure your mouse as left- or right-handed, specify how quickly you must click for the operating system to recognize a double-click, change the sensitivity of mouse motions, and set the shape and size of your system pointers. You can also select pointer schemes that change the pointer appearance based on the task being performed. Figure 5-12 shows the Mouse applet.



Scheme: Old Fashioned (system scheme)

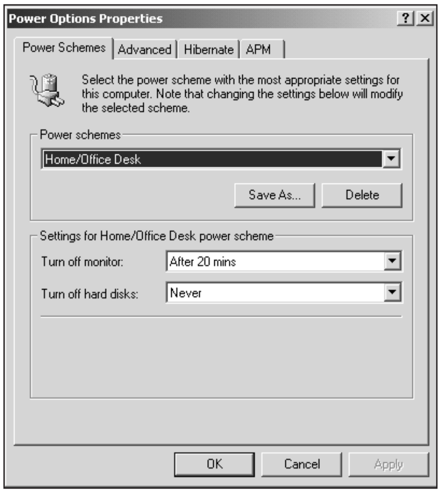
Customize:
- Normal Select
- Help Select
- Working In Background
- Busy
- Precision Select

☑ Enable pointer shadow

Figure 5-12 You can control mouse actions and pointer appearance through the Mouse applet

Power Management and Configuration

Another important applet in the Windows 2000 Control Panel is the Power Options applet. This applet enables you to configure the amount of time that must pass before the system "automagically" turns off the monitor or hard drive (see Figure 5-13). You can also enable "hibernation mode," which is the precursor to the OnNow system—it puts the computer into a very low-power suspended state after a set interval of inactivity.



Select the power scheme with the most appropriate settings for this computer. Note that changing the settings below will modify the selected scheme.

Power schemes: Home/Office Desk

Settings for Home/Office Desk power scheme

Turn off monitor: After 20 mins

Turn off hard disks: Never

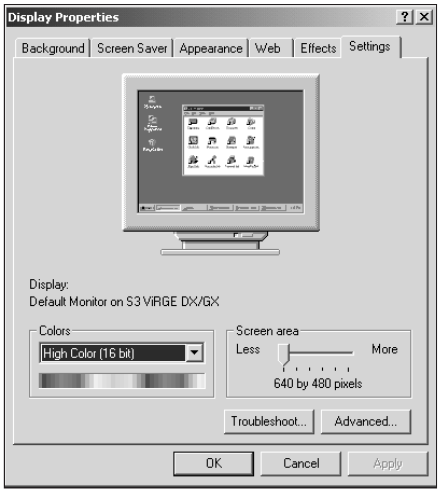
Figure 5-13 You can configure the many power management options of Win2K with the Power Options applet

Note

The appearance of the Power Options applet may differ on your computer based on its power manageability features. In the case of the Windows 2000 PC in Figure 5-13, the motherboard supports the older Advanced Power Management (APM) standard and you can enable APM features by clicking a check box. If you install Windows 2000 on a PC that does not support APM, you will not see this tab. Conversely, if you install Windows 2000 on a PC that supports ACPI management, you will see additional tabs and options in this applet.

Help Right Where You Need It

One of the most welcome additions to the Control Panel applets is a new Troubleshoot button, shown in Figure 5-14, which is located on the Settings tab of the Display applet.



Display:
Default Monitor on S3 ViRGE DX/GX

Colors: High Color (16 bit)

Screen area: Less — More
640 by 480 pixels

Figure 5-14 Notice the Troubleshoot button in the Display applet, Settings tab

You will also find a Troubleshoot button on all of the input device applets, the Phone and Modem applets, and the Scanners and Cameras applets. Clicking a Troubleshoot button takes you directly to the Windows 2000 Troubleshooting Wizard, which walks you through the troubleshooting process by asking a few simple questions that describe the type of problem(s) you are having and suggesting possible solutions. We'll talk about these new troubleshooting buttons at length in the next section.

What if It Breaks? Troubleshooting Devices in Windows 2000

As good as Windows 2000 is as an operating system, there is always the possibility that it might stop working correctly at some time. In particular, as you add and remove hardware from your PC, you might find yourself in the unenviable position of having to troubleshoot the hardware, drivers, and operating system configuration to resolve such problems. The bad news is that troubleshooting the complex combinations of hardware and software common to today's PCs remains more art than science. The good news is that Microsoft has given you invaluable ammunition in your troubleshooting battle in the form of expanded device help topics and new troubleshooting wizards. We'll examine the specific techniques you will use for troubleshooting Windows 2000 in detail, but first, let's look at some standard operating procedures for resolving hardware and driver problems.

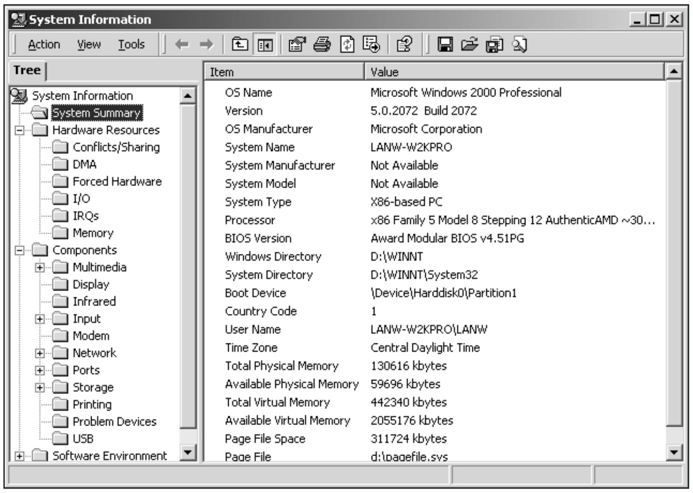
One Thing at a Time

The most often given—and most often ignored—advice for troubleshooting hardware is to change only one parameter, setting, or configuration option at a time whenever possible. When you adjust multiple variables at once, you have several possible outcomes—none of which is desirable: You might compound the problem instead of fixing it; you might accidentally fix the problem but have no idea which of your changes produced the favorable results; or you might do no harm and no good but lose track of your original settings along the way. All in all, it's best to stick to one change at a time. That is, if you suspect an IRQ conflict, make all possible IRQ changes to resolve the conflict *before* proceeding to I/O port changes or memory range changes.

Document, Document, Document

Carefully document your current configuration settings *before* beginning to make changes. Granted, documenting your configuration prior to making changes isn't really troubleshooting, but it is certainly a critical step in the troubleshooting process. The more cumulative changes you make, the more likely that you will not remember your original device settings when you try to undo all of your troubleshooting efforts.

The easiest way to document your computer settings prior to a foray into the wilds of troubleshooting is with the **MSInfo32** tool, also called the System Information tool. To access the System Information tool, select Start, Programs, Administrative Tools, Computer Management, expand the System Tools icon, and select System Information. As shown in Figure 5-15, the System Information utility contains a tree view in the left pane with which you can navigate among the four components of system information. Using this tool, you can print any component summary from the File menu.



Item	Value
OS Name	Microsoft Windows 2000 Professional
Version	5.0.2072 Build 2072
OS Manufacturer	Microsoft Corporation
System Name	LANW-W2KPRO
System Manufacturer	Not Available
System Model	Not Available
System Type	X86-based PC
Processor	x86 Family 5 Model 8 Stepping 12 AuthenticAMD ~30...
BIOS Version	Award Modular BIOS v4.51PG
Windows Directory	D:\WINNT
System Directory	D:\WINNT\System32
Boot Device	\Device\Harddisk0\Partition1
Country Code	1
User Name	LANW-W2KPRO\LANW
Time Zone	Central Daylight Time
Total Physical Memory	130616 kbytes
Available Physical Memory	59696 kbytes
Total Virtual Memory	442340 kbytes
Available Virtual Memory	2055176 kbytes
Page File Space	311724 kbytes
Page File	d:\pagefile.sys

Figure 5-15 The MSInfo32 utility offers a plethora of hardware and software configuration information

Another tool for documenting system configuration information is the Device Manager applet under the System utility in the Control Panel. To start the Device Manager, click Start, Settings, Control Panel, System, choose the Hardware tab, and then click the Device Manager button. Once in Device Manager, click the View menu and select **Print**. You can print this system summary on your default printer or send it to a text file.

Yet another option for documenting the system configuration prior to troubleshooting a hardware problem is to use third-party utilities. Many third-party utilities, such as the popular Norton Utilities, offer an option to document every detail and configuration setting for your Windows 2000 system.

General Troubleshooting Tips

The following is a list of general hardware troubleshooting steps for Windows 2000—or any operating system, for that matter:

Note

For additional information on troubleshooting Windows 2000, go to the online help for the operating system, or look at the Windows 2000 documentation. In addition, more detailed troubleshooting advice appears in Chapter 15.

- **How do I know that there is a problem?** If something isn't working correctly, it is usually good sign that a resource conflict, driver problem, or other anomaly is present. Always check the Device Manager in the Control Panel, System applet when you suspect a hardware problem. Any unresolved resource conflict will be prominently marked with a red X next to the device in question.
- **How do I resolve the problem?** You can modify some device settings in the Device Manager. Note that manual modifications of resource assignments should be used only as a last resort because of the automatic conflict resolution of Plug and Play hardware. One of your first steps should be to verify that you are using the latest device driver for every piece of hardware. You can download current drivers from most manufacturers' Web sites.
- **How can I avoid hardware problems to begin with?** Always be sure to use the latest device drivers for your hardware. Verify that your hardware is using the latest hardware BIOS updates. Many hardware manufacturers update the onboard code or BIOS for their hardware as problems are resolved or features are added.

Troubleshooting the Universal Serial Bus

By design, there are no user-configurable resource settings for USB devices. As a result, typical USB problems are usually related to physical connections. Be sure to comply with the cabling type and distance limitations and to set up the hub placement and hierarchical structure according to the USB specification. If a USB device is failing, try connecting the device to another known-good USB port. If the malfunctioning USB device is bus-powered, try connecting the device directly to a USB controller. Frequently, USB hubs do not support bus-powered devices that draw a large amount of current. As a last resort, you can delete the USB host controller under the Device Manager, restart your PC, and allow the operating system to redetect and reinstall the USB drivers. Also, be sure to use the latest drivers if your USB manufacturer offers specific drivers newer than those shipping with Windows 2000.

Troubleshooting the FireWire Bus

Similar to the USB bus and devices, the FireWire bus and devices should be self-configuring. In the event of FireWire-related problems, check your physical connections, cabling, and distance limitations. Updating FireWire drivers may also resolve FireWire-related problems.

Troubleshooting the PCI Bus

The PCI bus uses software-configurable hardware in lieu of dual inline package (DIP) switches and jumpers. The combination of PCI bus components and Plug and Play capabilities should prevent any hardware configuration and resource conflict problems. Some hardware permits you to set IRQ, I/O ports, media type, and memory address manually via manufacturer-provided software configuration utilities.

Troubleshooting Fatal Exception Errors

Typically, fatal exception errors are caused either by an incorrect device driver that does not correspond to the hardware installed or by an internal problem (such as a bug) related to poor device driver design. Either way, verifying that you have installed the latest device driver for that particular hardware will usually resolve fatal exception error problems.

CHAPTER SUMMARY

- Windows 2000 combines several advanced technologies to make hardware installation, configuration, and troubleshooting easier than ever before. The four components of hardware management in Windows 2000 work in unison to support the widest possible range of hardware while easing the burden for driver developers and system administrators. The combination of full Plug and Play support and ACPI specification support in Windows 2000 means that most hardware conflicts are a thing of the past, as are the demands on your time and the effort needed to resolve those conflicts. Extensive native support for many new hardware technologies and buses makes Windows 2000 the operating system of the future in many ways. As many large corporations turn their desktop attention to power and system management, Windows 2000's built-in management capabilities make the upgrade decision a snap.

- Microsoft has also spent considerable time and resources building troubleshooting help directly into the operating system. The System Information utility is an invaluable tool for reviewing and documenting your system configuration. Because of Microsoft's meticulous attention to the details of hardware support, Windows 2000 is without a doubt the most mature, well-documented, best-designed, and most extensively supported operating system ever released by the company.

Key Terms

Advanced Configuration and Power Interface (ACPI) specification — Defines Advanced Power Management features and is an integral part of the OnNow system built into Windows 2000. For more information, see the Microsoft Web site: *http://www.microsoft.com/hwdev/onnow/*.

Advanced Power Management (APM) — The legacy specification that implements power management in machine-specific BIOS code.

bus class driver — One of the native Windows 2000 driver layers, which provides all basic driver functionality for bus devices.

bus minidriver — A small device driver that implements manufacturer-specific features not included in the bus class driver. It works in conjunction with the bus class driver.

CardBus — A high-speed bus specification based on the PCMCIA technology found on laptop computers. This hardware interface supports PC Card peripheral technologies.

device class driver — A layer of built-in device support that implements basic support for a class of hardware, such as modems. A device class driver supports all generic or standard features of a particular type of peripheral, thereby easing the development burden for hardware manufacturers.

device driver interfaces (DDIs) — Interfaces that define how device drivers interact with the operating system components, such as OnNow.

Device Manager — An internal Windows 2000 device management routine that handles enumeration, Plug and Play configuration, and device support.

device minidriver — A small device driver that implements manufacturer-specific features not included in the device class driver. It works in conjunction with the device class driver.

driver stack — The entire device driver layer in Windows 2000, including the HAL, bus class and minidrivers, and device class and minidrivers.

enumeration — The process by which Plug and Play adapters are recognized by the operating system and a device tree is built.

hardware devices — Physical hardware, features, and interfaces installed in a PC.

human interface device class — One of the Windows 2000 driver classes devoted to handling input devices such as mice, keyboards, and game controllers.

IEEE 1394 Serial Bus (FireWire) — A high-speed serial bus that supports 63 devices per bus, allows interconnection of 1023 buses, and features automatic device recognition.

legacy virtualization drivers layer — A layer in the driver stack that supports legacy VxD-style device drivers.

MSInfo32 — A system configuration and documentation utility that reports numerous hardware and software settings. Also called the System Information tool.

multiple display support — Native support within Windows 2000 that allows definition and use of as many as nine display monitors.

OnNow system — A Microsoft specification that supports hibernation, "instant-on," and sophisticated power management features. For more information, see the Microsoft Web site: *http://www.microsoft.com/hwdev/onnow/*.

PC Card — Laptop peripheral technology based on the CardBus specification. Similar in design to PCMCIA cards but operating at a higher bus speed.

PCI bus — High-performance personal computer bus that allows component-to-component communication without the need for CPU intervention.

Plug and Play — A hardware specification that allows automatic discovery and configuration of hardware devices.

Plug and Play Manager — The Windows 2000 component that handles operating system recognition of Plug and Play hardware.

power management — The Windows 2000 component that provides operating system power management features and controls hardware power management features.

system buses — The Windows 2000 component that recognizes and controls system buses such as PCI, CardBus, FireWire, and USB.

Universal Serial Bus (USB) — A new high-speed serial bus that supports 127 peripheral devices and automatic device configuration.

Virtual DOS Machine — A software environment within Windows 2000 that supports legacy DOS programs running in a protected environment space.

VxD driver — The legacy device driver model, still supported under Windows 2000, that requires much more development effort than corresponding WDM drivers.

Win32 Driver Model (WDM) — The new Windows driver model that allows simplified device driver development such that one driver can be used on both Windows 2000 and Windows 98 systems.

REVIEW QUESTIONS

1. There are ________________ components of hardware management in Windows 2000.
2. What is the acronym for the new Microsoft driver model developed for Windows 2000 and Windows 98?
   a. WfM
   b. DPMI
   c. WDM
   d. DMI
3. The hardware abstraction layer permits applications to directly access system hardware. True or False?
4. Enumeration is the process Windows 2000 uses to do which of the following?
   a. Build the device tree
   b. Perform internal operating system math computations
   c. Perform floating-point calculations
   d. Recognize Plug and Play hardware and resources in use

5. The ________________ bus allows high-bandwidth component-to-component communication without CPU intervention.
6. The OnNow system allows a user to immediately do which of the following?
   a. Boot the computer from a hibernation state
   b. Find a missing PC anywhere on the network
   c. Manage the power requirements of a Windows 2000 PC
   d. Trace data flows between system components
7. The device driver stack includes which of the following? (Choose all the correct answers.)
   a. HAL
   b. PCI bus
   c. Device class drivers and minidrivers
   d. System RAM
   e. Bus class drivers and minidrivers
   f. The legacy virtualization drivers layer
   g. External CPU cache
8. APM is the latest development in power management for personal computers. True or False?
9. VxD is the recommended driver model for hardware manufacturers to follow when developing device drivers for use under Windows 2000. True or False?
10. ________________ is the high-speed bus that supports PC Cards in laptop computers.
11. You can view and print system hardware and software settings using which of the following utilities? (Choose all correct answers.)
    a. ViewConf
    b. MSInfo32
    c. Device Manager
    d. Services applet in the Control Panel
12. The multiple display support in Windows 2000 supports as many as ________________ displays.
13. The IEEE 1394 Serial Bus is also known as which of the following?
    a. FastBus
    b. WireFire
    c. USB
    d. FireWire

14. The ACPI specification includes which of the following features? (Choose all correct answers.)
    a. Advanced operating system-directed power management
    b. The ability to turn peripherals on and off as needed
    c. The ability to manage data flow between system components
    d. The OnNow system
15. USB supports a maximum of how many devices?
    a. 127 devices
    b. 63 devices
    c. 1023 devices
    d. 63 devices per bus and 1023 buses
16. What is the first step in troubleshooting hardware?
    a. Rebooting
    b. Unplugging the PC
    c. Documenting the system configuration
    d. Washing your hands
17. Hardware resource conflicts are most readily visible in the ________________ applet in the Control Panel.
18. ________________ and ________________ is the standard for automatic recognition and resolution of hardware resource conflicts.
19. A device minidriver is a very complex and difficult-to-develop device driver written for use under Windows 2000. True or False?
20. A bus class driver contains generic and standardized features common to all devices that utilize a specific bus. True or False?
21. Device class drivers are provided by device manufacturers, whereas device minidrivers are included in Windows 2000. True or False?
22. The human interface device class controls user ergonomics and screen appearance. True or False?
23. Device drivers developed for use under Win95 work properly under Windows 2000 because of the legacy virtualization drivers layer. True or False?
24. Which of the following buses is supported under Windows 2000? (Choose all correct answers.)
    a. PCI
    b. USB
    c. BART
    d. CardBus
    e. FireWire

25. ________________ is the acronym for interfaces that define how device drivers interact with operating system components such as OnNow.
26. The ACPI specification is implemented solely in the system BIOS. True or False?
27. The Virtual DOS Machine support in Windows 2000 supports legacy DOS programs in a protected environment space. True or False?
28. With respect to Windows 2000, what does hibernation refer to?
    a. Suspending operation of a system component or peripheral
    b. Variable-speed CPU support
    c. Variable-bandwidth allocation for network operations
29. What is the best way to avoid fatal exception errors?
    a. Download and install Windows 2000 Service Pack 14
    b. Never turn off the PC without properly shutting it down
    c. Always use the latest device drivers for your installed hardware
30. What is the best way to resolve hardware problems?
    a. Change many parameters at once in an effort to expedite the troubleshooting process
    b. Document your configuration first and change only one parameter at a time whenever possible
    c. Reinstall the operating system
    d. Avoid hardware problems to begin with by upgrading your PC

HANDS-ON PROJECTS

Hands-On Project

Project 5-1

The most important task you can perform now that will aid you in all future hardware troubleshooting efforts is to document your system setup. Although you can retrieve, view and print system configuration information in several different utilities, you might want to use the MSInfo32 utility to print a complete list of all hardware configurations.

MSInfo32 tallies hardware IRQs in use, memory address usage, I/O ports in use, and software components. You should define a MSInfo32 icon on your desktop to ease the use of this utility and to provide a constant reminder of the importance of system documentation. To create such an icon:

1. Right-click on your desktop and select **New**, **Shortcut** from the menu.
2. Click the **Browse** button and follow the path to **%System_Drive%\Program Files\Common Files\Microsoft Shared\MSInfo**.
3. Click **Msinfo32.exe**, and then click **OK**. See Figure 5-16.

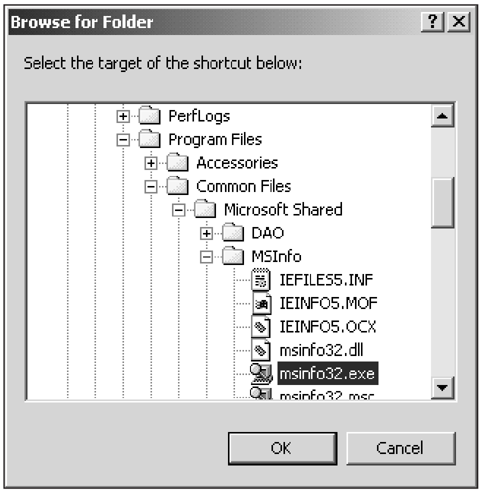


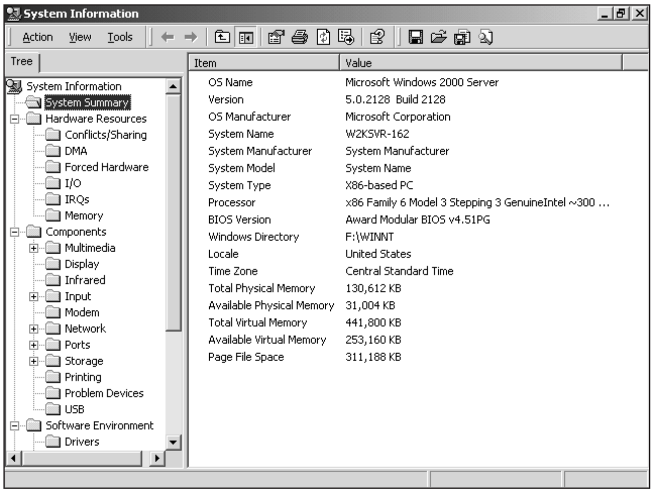
Figure 5-16 Creating a shortcut to Msinfo32.exe

4. Click the **Next** button, then click the **Finish** button. You should now have an MSInfo32 shortcut icon on your desktop.

Project 5-2

Now that you have a shortcut MSInfo32 icon on your desktop, follow these steps to document your system configuration:

1. Double-click the **Msinfo32.exe** icon to launch the Msinfo32 utility. See Figure 5-17.



Item	Value
OS Name	Microsoft Windows 2000 Server
Version	5.0.2128 Build 2128
OS Manufacturer	Microsoft Corporation
System Name	W2KSVR-162
System Manufacturer	System Manufacturer
System Model	System Name
System Type	X86-based PC
Processor	x86 Family 6 Model 3 Stepping 3 GenuineIntel ~300 ...
BIOS Version	Award Modular BIOS v4.51PG
Windows Directory	F:\WINNT
Locale	United States
Time Zone	Central Standard Time
Total Physical Memory	130,612 KB
Available Physical Memory	31,004 KB
Total Virtual Memory	441,800 KB
Available Virtual Memory	253,160 KB
Page File Space	311,188 KB

Figure 5-17 The MSInfo32 utility provides a complete list of system information

2. Click the **Action** menu.

3. Select **Print** from the Action menu. You can print to a text file (set up a text file output device in the **Printers** applet) or to one of the attached printers.
4. Select **OK**, and your system configuration will be printed on the selected printer.

Project 5-3

You can also save your system configuration to a file format called a System Information File. If you ever contact Microsoft or a third-party technical support provider, someone might ask you to save your system configuration information to a System Information File and e-mail the file to the company. To save your system configuration to a System Information File:

1. Start the MSInfo32 utility by choosing **Start**, **Programs**, **Administrative Tools**, **Computer Management**, expanding the **System Tools** icon, and selecting **System Information**.
2. From the **Action** menu, select **Save As System Information File**.
3. Enter a descriptive filename—for example, lanwserver, with no file extension (.nfo is automatically appended to the filename).
4. Click **Save**, and wait for the file to be saved.
5. If you need to e-mail the file, you can find the saved configuration file in the **My Documents** folder unless you specify another destination folder as you save the file.

Case Projects

1. You have just installed a new piece of Plug and Play hardware in your computer. After the hardware installation is complete, you restart the computer. What steps must be taken for Windows 2000 to recognize the newly installed device? Will you have to manually configure resources for the new hardware? If so, how will you configure those resources? If you choose to update the driver for this device in the future, what is the easiest way to do so?
2. Your modem stops working after you install a new serial I/O adapter in your PC. You notice a red X next to your modem device in the Device Manager. How will you discover what resource conflict is causing the malfunction? If you do not know how to resolve the conflict, where can you turn for help? Are there any "gotchas" to be aware of when manually adjusting hardware resources in the Device Manager?
3. You've just been assigned the task of training a new employee who will be helping with hardware support within your organization. Describe the salient points you must communicate to the trainee regarding troubleshooting hardware devices under Windows 2000.
4. Describe the difference between hardware and devices as seen by Windows 2000 and how each are managed or handled under the Windows 2000 operating system.